AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1. (Currently Amended) A data dependent scrambler for a communications channel that receives a user data sequence with a plurality of symbols, comprising:

a seed finder that generates a scrambling seed that is dependent upon said symbols in said user data sequence;

a first scrambler that receives said user data sequence and said scrambling seed and that generates ~~said~~ a scrambled user data sequence;

a code finder that generates at least one of a token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

an encoder that receives said scrambled user data sequence and at least one of said token and said offset and that increases a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

2. (Original) The data dependent scrambler of Claim 1 wherein at least one of said symbols of said user data sequence includes dummy bits and wherein said encoder stores said offset in said dummy bits.

3. (Original) The data dependent scrambler of Claim 2 wherein said user data sequence includes Y dummy bits, said symbols include M bits, said user data sequence includes X bits, and Y is equal to M minus a remainder of X divided by M.

4. (Original) The data dependent scrambler of Claim 1 further comprising a data buffer that stores said user data sequence until said scrambling seed is generated.

5. (Original) The data dependent scrambler of Claim 1 wherein said data dependent scrambler receives said user data sequence.

6. (Original) The data dependent scrambler of Claim 1 further comprising an ECC/CRC encoder that encodes said scrambled user data sequence and that appends ECC and CRC bits to said scrambled user data sequence.

7. (Original) The data dependent scrambler of Claim 6 wherein said ECC/CRC encoder outputs said scrambled user data sequence to a run length limited (RLL) encoder that generates a RLL sequence that is based on said ECC and CRC bits.

8. (Original) The data dependent scrambler of Claim 1 further comprising an interleaving encoder that receives said scrambled user data sequence from said encoder and that reduces the number of consecutive zeros in interleaved subsequences of said scrambled user data.

9. (Original) The data dependent scrambler of Claim 1 wherein said data dependent scrambler is implemented in a write path of a data storage system.

10. (Original) The data dependent scrambler of Claim 1 wherein said first scrambler performs a bitwise exclusive (XOR) operation.

11. (Currently Amended) A communications channel that receives a user data sequence including a plurality of symbols, comprising:

a host bus interface (HBI) that transmits said user data sequence including a plurality of symbols; and

a data dependent scrambler that communicates with said HBI, that scrambles said user data sequence using a scrambling seed that is data dependent, that increases a Hamming weight of said scrambled user data sequence using a token, that calculates an offset between said scrambling seed and said token, and that stores said offset of dummy bits of at least one symbol of said scrambled user data sequence.

12. (Original) The communications channel of Claim 11 wherein said data dependent scrambler includes:

a seed finder that generates said scrambling seed that is dependent upon said symbols in said user data sequence; and

a first scrambler that receives said user data sequence and said scrambling seed and that generates said scrambled user data sequence.

13. (Original) The communications channel of Claim 12 wherein said data dependent scrambler further includes:

a token finder that generates at least one of said token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

an encoder that receives said scrambled user data sequence and said at least one of said token and said offset and that increases a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

14. (Original) The communications channel of Claim 11 wherein said user data sequence includes X bits organized as N M-bit symbols, wherein at least one symbol includes Y dummy bits, and wherein Y is equal to M minus a remainder of X divided by M.

15. (Original) The communications channel of Claim 11 wherein said data dependent scrambler further includes a data buffer that stores said user data sequence until said scrambling seed is generated.

16. (Original) The communications channel of Claim 11 further comprising:

a buffer manager that receives said user data sequence from said HBI;

a buffer that communicates with said buffer manager; and

a disk formatter that communicates with said buffer manager.

17. (Original) The communications channel of Claim 11 further comprising an error correction coding (ECC)/CRC encoder that encodes said scrambled user data sequence and that appends ECC and CRC bits to said scrambled user data sequence.

18. (Original) The communications channel of Claim 18 further comprising a run length limited (RLL) encoder that generates a RLL sequence that is based on said ECC and CRC bits.

19. (Original) The communications channel of Claim 11 further comprising an interleaving encoder that receives said scrambled user data sequence from said encoder and that reduces the number of consecutive zeros in interleaved subsequences of said scrambled user data.

20. (Original) The communications channel of Claim 11 wherein said data dependent scrambler is implemented in a write path of a data storage system.

21. (Original) The communications channel of Claim 11 wherein said first scrambler perform a bitwise exclusive (XOR) operation.

22. (Currently Amended) A data dependent scrambler for a communications channel that receives a user data sequence including a plurality of symbols, comprising:

seed finding means for generating a scrambling seed that is dependent upon said symbols in said user data sequence;

first scrambling means for receiving said user data sequence and said scrambling seed and for generating ~~said~~ a scrambled user data sequence;

token finding means for generating at least one of a token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

first encoding means for receiving said scrambled user data sequence and said at least one of said token and said offset and that increases a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

23. (Original) The data dependent scrambler of Claim 22 wherein at least one of said symbols of said user data sequence includes dummy bits and wherein said first encoding means stores said offset in said dummy bits.

24. (Original) The data dependent scrambler of Claim 23 wherein said user data sequence includes X bits arranged in M-bit symbols, said user data sequence includes Y dummy bits and Y is equal to M minus a remainder of X divided by M.

25. (Original) The data dependent scrambler of Claim 22 further comprising data buffer means for storing said user data sequence while said scrambling seed is generated.

26. (Original) The data dependent scrambler of Claim 22 further comprising second encoding means for encoding said scrambled user data sequence and for appending ECC and CRC bits to said scrambled user data sequence.

27. (Original) The data dependent scrambler of Claim 26 wherein said second encoding means outputs said scrambled user data sequence to a third encoding means for generating an RLL sequence that is based on said ECC and CRC bits.

28. (Original) The data dependent scrambler of Claim 22 further comprising interleave encoding means for receiving said scrambled user data sequence from said first encoding means and for reducing the number of consecutive zeros in interleaved subsequences of said scrambled user data.

29. (Original) The data dependent scrambler of Claim 22 wherein said data dependent scrambler is implemented in a write path of a data storage system.

30. (Original) The data dependent scrambler of Claim 22 wherein said first scrambling means performs a bitwise exclusive (XOR) operation.

31. (Currently Amended) A communications channel that receives a user data sequence including a plurality of symbols, comprising:

interface means for transmitting said user data sequence including a plurality of symbols; and

data dependent scrambling means for communicating with said interface means, for scrambling said user data sequence using a scrambling seed that is data dependent, for increasing a Hamming weight of said scrambled user data sequence using a token, for calculating an offset between said scrambling seed and seed token, and for storing said offset in dummy bits in at least one symbol of said scrambled user data sequence.

32. (Original) The communications channel of Claim 31 wherein said data dependent scrambling means includes:

seed finding means for generating said scrambling seed that is dependent upon said symbols in said user data sequence; and

first scrambling means that receives said user data sequence and said scrambling seed and that generates said scrambled user data sequence.

33. (Original) The communications channel of Claim 32 wherein said data dependent scrambling means further includes:

token finding means for generating at least one of said token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

first encoding means for receiving said scrambled user data sequence and at least one of said token and said offset and for increasing a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

34. (Original) The communications channel of Claim 33 wherein at least one of said symbols of said user data sequence includes dummy bits and wherein said first encoding means stores said offset in said dummy bits.

35. (Original) The communications channel of Claim 31 wherein said user data sequence includes X bits organized as N M-bit symbols, wherein at least one symbol includes Y dummy bits, and wherein Y is equal to M minus a remainder of X divided by M.

36. (Original) The communications channel of Claim 31 wherein said data dependent scrambling means further includes data buffer means for storing said user data sequence while said scrambling seed is generated.

37. (Original) The communications channel of Claim 31 further comprising:

buffer managing means for receiving said user data sequence from said interface means;

buffer means for communicating with said buffer managing means; and

disk formatting means for communicating with said buffer managing means.

38. (Original) The communications channel of Claim 31 further comprising second encoding means for encoding said scrambled user data sequence and for appending ECC and CRC bits to said scrambled user data sequence.

39. (Original) The communications channel of Claim 38 further comprising third encoding means for generating an RLL sequence that is based on said ECC and CRC bits.

40. (Original) The communications channel of Claim 31 further comprising interleave encoding means for receiving said scrambled user data sequence from said first encoding means and for reducing the number of consecutive zeros in interleaved subsequences of said scrambled user data.

41. (Original) The communications channel of Claim 31 wherein said communications channel is a write path of a data storage system.

42. (Original) The communications channel of Claim 31 wherein said first scrambling means performs a bitwise exclusive (XOR) operation.

43. (Original) A method of scrambling a user data sequence including a plurality of symbols, comprising:

generating a scrambling seed that is dependent upon said symbols in said user data sequence;

scrambling said user data sequence to provide a scrambled user data sequence based on said scrambling seed;

generating at least one of a token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

increasing a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

44. (Original) The method of Claim 43 further comprising storing said offset in dummy bits of said user data sequence.

45. (Original) The method of Claim 44 wherein said user data sequence includes X bits arranged as N M-bit symbols and said user data sequence includes Y dummy bits and wherein Y is equal to M minus a remainder of X divided by M.

46. (Original) The method of Claim 43 further comprising storing said user data sequence while said scrambling seed is generated.

47. (Original) The method of Claim 43 further comprising:

encoding said scrambled user data sequence with an error correction coding (ECC) and cyclic redundancy check (CRC) bits; and

appending ECC and CRC bits to said scrambled user data sequence.

48. (Original) The method of Claim 47 further comprising generating an RLL sequence that is based on said ECC and CRC bits.

49. (Original) The method of Claim 43 further comprising interleaving said scrambled user data sequence to reduce the number of consecutive zeros in interleaved subsequences of said scrambled user data.

50. (Original) The method of Claim 43 wherein said scrambling step includes performing a bitwise exclusive (XOR) operation.

51. (Currently Amended) A method of processing a user data sequence including a plurality of symbols through a communications channel, comprising:

transmitting said user data sequence including a plurality of symbols to a data dependent scrambler through an interface;

scrambling said user data sequence using a scrambling seed that is data dependent;

increasing a Hamming weight of said scrambled user data sequence using a token;

calculating an offset between said scrambling seed and said token; and

storing said offset in dummy bits in at least one symbol of said scrambled user data sequence.

52. (Original) The method of Claim 51 further comprising:

generating said scrambling seed based on said symbols in said user data sequence; and

receiving said user data sequence and said scrambling seed; and

generating said scrambled user data sequence.

53. (Original) The method of Claim 52 further comprising:

generating at least one of said token that is dependent upon said symbols in said user data sequence and an offset of said token from said scrambling seed; and

receiving said scrambled user data sequence and at least one of said token and said offset; and

increasing a Hamming weight of said scrambled user data sequence using said at least one of said token and said offset.

54. (Original) The method of Claim 53 further comprising storing said offset in dummy bits of at least one of said symbols of said user data sequence.

55. (Original) The method of Claim 51 wherein said user data sequence includes X bits organized as N M-bit symbols, wherein at least one symbol includes Y dummy bits, and wherein Y is equal to M minus a remainder of X divided by M.

56. (Original) The method of Claim 51 further comprising storing said user data sequence while said scrambling seed is generated.

57. (Original) The method of Claim 51 further comprising:

encoding said scrambled user data sequence with an error correction coding (ECC) and cyclic redundancy check (CRC) bits; and

appending ECC and CRC bits to said scrambled user data sequence.

58. (Original) The method of Claim 57 further comprising generating a run length limited (RLL) sequence that is based on said ECC and CRC bits.

59. (Original) The method of Claim 51 further comprising interleaving said scrambled user data sequence to reduce the number of consecutive zeros in interleaved subsequences of said scrambled user data.

60. (Original) The method of Claim 51 wherein said scrambling step includes performing a bitwise exclusive (XOR) operation.